

# Pathway Prediction Using Similar Users and the N-gram Model

Kanta Kawase

Intelligent Computer Entertainment Laboratory  
Graduate School of Information Science and Engineering  
Ritsumeikan University  
1-1-1 Nojihigashi, Kusatsu, Shiga 525-8577, Japan  
is0011ps@ed.ritsumei.ac.jp

Ruck Thawonmas

Intelligent Computer Entertainment Laboratory  
Department of Human and Computer Intelligence  
Ritsumeikan University  
1-1-1 Nojihigashi, Kusatsu, Shiga 525-8577, Japan  
ruck@ci.ritsumei.ac.jp

**Abstract**—This paper is about our research on user pathway prediction for being applied to a location aware system. In particular, we propose a prediction method based on an  $N$ -gram model with Kneser-Ney smoothing (KNS), originally developed by other researchers for statistical language model smoothing, and introduce the use of the transition information of similar users into KNS. We then verify the performance of the proposed prediction method by comparing it with an existing prediction method and a prediction method based on KNS using all users' information. The comparison result reveals that the proposed method outperforms its counterparts on all performance metrics: precision, recall,  $F$ -measure, and CA.

**Keywords**—pathway prediction; kneser-ney smoothing;  $N$ -gram Model

## I. INTRODUCTION

Recently, location information measurement technologies and wireless communication technologies have been developed and spread. This brings about environments where one can obtain information regardless of their location and time. From this background, location-aware (or context-aware) systems [1],[2] exist that provide services to their users in optimal timing using the users' position information.

Such services include presenting the users with the transportation information towards their traveling destinations or the information on stores located there. In these kinds of services, the users' time and load can be reduced if their destinations can be predicted and shown without direct inputs from the users. Hence, prediction of the users' movement places with a high accuracy is important.

Previous methods for pathway prediction include the work by Bohnert and Zukerman [3] and that of Cheng et al. [4]. Bohnert and Zukerman proposed a method based on movement transitions. They introduced the similar movement information into the transition probabilities. However, some transitions might have missing values. To complement such missing values, Cheng et al. proposed a method based on matrix factorization. However, their method does not exploit the information from users with similar movements. In addition, both methods have also some parameters that need to

be carefully determined. These issues are coped with in our proposed method.

In this paper, we propose and verify a highly precise movement place prediction method with the aim of applying it to a location-aware system. An example of such a system is a location-aware game, under development at the authors' laboratory, for use in a museum. The contributions of this paper are as follows:

- 1) The first application, to our knowledge, of Kneser-Ney smoothing (KNS), a high precision smoothing technique for prediction of words in the natural language-processing field, to user pathway prediction.
- 2) Our mechanism for incorporation of the information from similar users into KNS.
- 3) The proposed prediction method that outperforms other prediction methods on all performance metrics: precision, recall,  $F$ -measure, and CA.

In the rest of this paper, an outline of an  $N$ -gram model and smoothing techniques including KNS are discussed in Section 2, followed by the proposed method in Sections 3. Our experiment for verifying the effectiveness of the proposed method is given in Section 4, followed by conclusions and future work.

## II. N-GRAM MODEL AND SMOOTHING TECHNIQUES

### A. $N$ -gram Model

An  $N$ -gram model [5] is a model, in natural language processing, for calculating the probability of generating a word of interest using the information on the last  $(N-1)$  words. In this work, we replace a word and its history as a place visited by a user of interest and his/her visit-history and predict the user's next visiting place by an  $N$ -gram model. Assuming the  $(N-1)$ th Markov property of a word string (word history)  $w_1^n = w_1, \dots, w_n$  the generation probabilities can be given by

$$p(w_1^n) = \prod_{i=1}^n p(w_i | w_1^{i-1}) \approx \prod_{i=1}^n p(w_i | w_{i-N+1}^{i-1}) \quad (1)$$

The probability  $p(w_i | w_{i-N+1}^{i-1})$  in Eqn. 1 can be derived with maximum likelihood estimation as shown in Eqn. 2 by the relative frequency of the word string.

$$p(w_i | w_{i-N+1}^{i-1}) = \frac{c(w_{i-N+1}^i)}{c(w_{i-N+1}^{i-1})} \quad (2)$$

where  $w_i$  represents a candidate predicted word, and  $c(w_{i-N+1}^i)$  represents the appearance frequency of the word sequence  $w_{i-N+1}^i$ . In the case of  $N=1, 2,$  and  $3,$  it is called *unigram*, *bigram*, *trigram*, respectively. The model that assumes every word occurs with an equal probability is called *zerogram*.

### B. $N$ -gram Smoothing

In an  $N$ -gram model,  $X^{(N-1)}$  states exist for  $X$ : the total number of the words. As a result, the amount of training data required for reliable estimation grows exponentially with  $N$ . When performing maximum likelihood estimation in such a case, some states (word sequences) might not have supporting data in the training data set, causing their probabilities to become 0. In addition, the accuracy in predicting words with low occurrence frequency is impaired. This problem is called the zero frequency problem or sparseness problem.

To address the above problem of  $N$ -gram, techniques called smoothing exist that adjust zero or low probabilities – those with zero or low relative frequency – upward and high probabilities – those with high relative frequency – downward. Smoothing thus leads probability distributions to be more uniform, resulting in more prediction accuracy.

For a  $N$ -gram model, a technique called *absolute discounting* [5] decreases the probability of each observed the higher order model by a constant  $D$  and equally redistribute  $D$  to the lower-order model as

$$p_{abs}(w_i | w_{i-n+1}^{i-1}) = \frac{\max(c(w_{i-n+1}^i) - D, 0)}{\sum_{w_i} c(w_{i-n+1}^{i-1})} + \frac{D}{\sum_{w_i} c(w_{i-n+1}^{i-1})} N_{1+}(w_{i-n+1}^{i-1} \cdot) p_{abs}(w_i | w_{i-n+2}^{i-1}) \quad (3)$$

where  $N_{1+}(w_{i-n+1}^{i-1} \cdot)$  represents the number of  $N$ -grams starting with  $w_{i-n+1}^{i-1}$ . Due to smoothing, a suitable probability can be obtained for each state which does not exist or exists with low relative frequencies in the training data.

A more precise smoothing technique is KNS [5], and as mentioned earlier we use KNS in our work. KNS equation is given as follows

$$p_{KN}(w_i | w_{i-n+1}^{i-1}) = \frac{\max(c(w_{i-n+1}^i) - D, 0)}{\sum_{w_i} c(w_{i-n+1}^{i-1})} + \frac{D}{\sum_{w_i} c(w_{i-n+1}^{i-1})} N_{1+}(w_{i-n+1}^{i-1} \cdot) p_{KN}(w_i | w_{i-n+2}^{i-1}) \quad (4)$$

KNS's lower-order model in Eqn. 4 is given as follows

$$p_{KN}(w_i | w_{i-n+1}^{i-1}) = \frac{N_{1+}(\cdot w_{i-n+2}^i)}{N_{1+}(\cdot w_{i-n+2}^{i-1})} \quad (5)$$

where the numerator represents the number of unique words that appear before  $w_{i-n+2}^i$  in the training data while the denominator represents the combination of words that appear before  $w_{i-n+2}^{i-1}$  and words that appear after  $w_{i-n+2}^{i-1}$ .

In the rest of this paper, we base our work on a bigram model. The bigram KNS equation is thereby

$$p_{KN}(w_i | w_{i-1}) = \frac{\max(c(w_{i-1}^i) - D, 0)}{c(w_{i-1})} + \frac{D}{c(w_{i-1})} N_{1+}(w_{i-1} \cdot) p_{KN}(w_i) \quad (6)$$

KNS estimates the probability of a unigram as follows

$$p_{KN}(w_i) = \frac{N_{1+}(\cdot w_i)}{N_{1+}(\cdot)} \quad (7)$$

where the numerator represents the number of unique words that appear before  $w_i$  in the training data while the denominator represents the number of appeared bigram types. In addition,  $D$  is usually set to

$$D = \frac{n_1}{n_1 + 2 \cdot n_2} \quad (8)$$

where  $n_1$  and  $n_2$  are the number of bigram types appearing once and of those appearing twice in the training data, respectively. Note that if  $n_1$  is large, the value of  $D$  will be close to its maximum, decreasing the influence of the first term and increasing that of the second term in Eqn. 6, and vice versa.

## III. PROPOSED METHODOLOGY

Let  $a$  and  $A$  denote a targeted user and training users, respectively. Our first algorithm for pathway prediction by KNS is shown below.

---

### Algorithm 1 Pathway Prediction by KNS

---

- 1: Acquire information such as the number of occurrences of transitions from both  $a$  and  $A$ .
  - 2: Calculate the transition probability to each candidate place from the present location of  $a$  by KNS.
  - 3: Set the next place as the one with the highest transition probability from the present location of  $a$ .
-

We improve the above algorithm based on our thought that users having similar stay time on a same place also have similar movement preference. In the improved algorithm, only information on similar users in  $A$ , those with similarity to  $a$  above a given threshold, is used, and this algorithm is given below.

---

**Algorithm 2** Pathway Prediction by Similar User KNS

---

- 1: Extract users  $S$  that are similar to  $a$  from  $A$ . (Fig. 1)
  - 2: Acquire information such as the number of occurrences of transitions from both  $a$  and  $S$ .
  - 3: Calculate the transition probability to each candidate place from the present location of  $a$  by KNS.
  - 4: Set the next place as the one with the highest transition probability from the present location of  $a$ .
- 

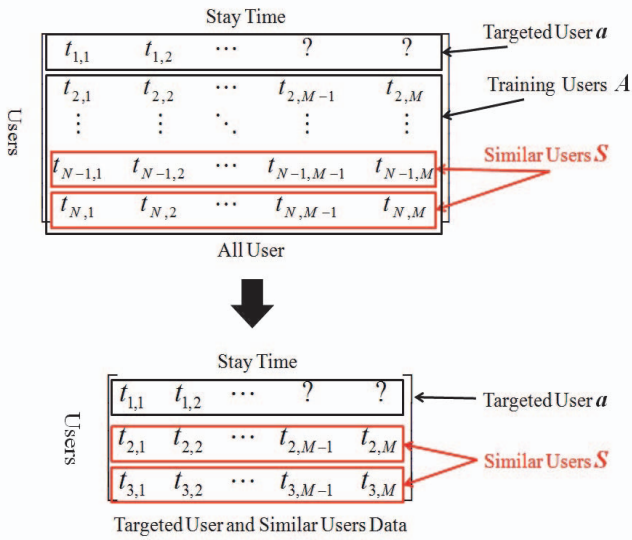


Fig.1. Extraction of similar users

---

**Algorithm 3** Threshold Determination

---

- 1: **for** each threshold value in  $[0.1, 0.9]$  with the interval of 0.1 **do**
  - 2:     **for** each user in  $A$  **do**
  - 3:         Divide  $A$  into the current user  $b$  (validated user) and new training users  $B$ .
  - 4:         Extract users  $S'$  from  $B$  with the stay-time similarity to  $b$  higher than the current threshold.
  - 5:         Acquire information such as the number of occurrences of transitions from both  $b$  and  $S'$ .
  - 6:         Calculate the transition probability to each candidate place from the present location of  $b$  by KNS.
  - 7:         Obtain the prediction accuracy (with F-measure) for  $b$ .
  - 8:     **end for**
  - 9:     Calculate the average prediction accuracy among all validated users at the current threshold value.
  - 10: **end for**
  - 11: Select the threshold value with the highest average prediction accuracy.
- 

The threshold to extract users similar to  $a$  is determined according to the following algorithm (Fig. 2).

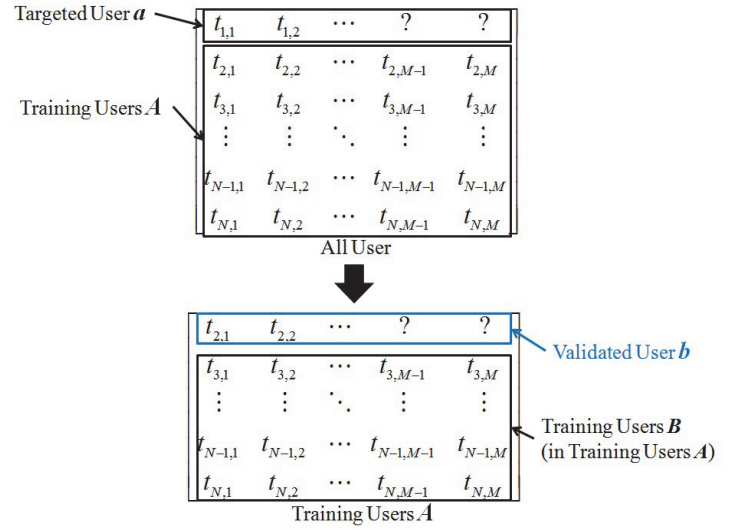


Fig.2. User division for threshold determination

The stay-time similarity between users  $u$  and  $v$  is calculated using either the cosine similarity (Eqn. 9) or the reverse of Euclidean distance (Eqn. 10).

$$sim_{\cos}(u, v) = \frac{\sum_i t_{ui} t_{vi}}{\sqrt{\sum_i t_{ui}^2} \sqrt{\sum_i t_{vi}^2}}, \quad (9)$$

$$sim_{euclid}(u, v) = \frac{1}{\sqrt{\sum_i (t_{ui} - t_{vi})^2}}, \quad (10)$$

where  $t_i$  is  $*$ 's stay time at visited place  $i$ .

#### IV. EXPERIMENT

We verified the prediction accuracy of our method using exhibits browsing data in the MIT Museum<sub>1</sub>, used in existing work [6] and [7]. This set of data contains visit history of 45 users' view time at 12 in the museum. Data were obtained by asking those users to voluntarily view exhibits therein. In our experiment, when the stay time of user  $u$  at exhibit  $i$  is not 0, we assumed that  $u$  visited and viewed  $i$ . In addition, the users in this data set only moved in one direction from the entrance to the exit. The layout of the museum is shown in Fig.3.

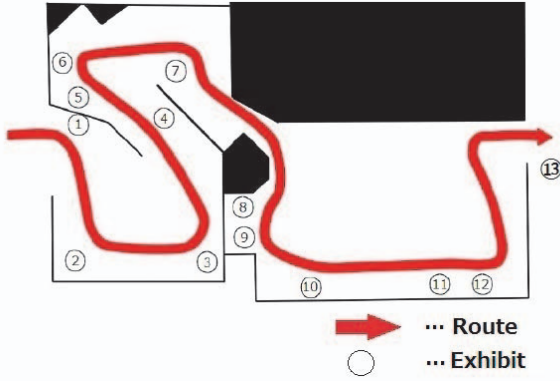


Fig.3. The layout of the MIT Museum, displaying 12 exhibits, a visit data set of which is used in this work

In our experiment, we modified the data by considering the exit as the terminal virtual exhibit 13, not present in the original data (user data 1). A part of the modified data is shown in the table I. In addition, another experiment was performed using the reversed data (user data 2) of the original data.

TABLE I. EXAMPLE OF USER DATA1 IN THE EXPERIMENT (EXH FOR EXHIBIT)

	Exh 1	Exh 2	Exh 3	...	Exh 12	Exh 13
User 1	25	0	41	...	30	1
User 2	5	10	5	...	7	1
User 3	0	5	0	...	0	1
User 44	30	5	0	...	45	1
User 45	15	5	0	...	10	1

#### A. Overall Visit and Progressive Visit

We based our experiment on paper the experiment in paper [3], where *Overall Visit (OV)* and *Progressive Visit (PV)* were conducted.

*OV* is described as follows:

- 1) Divide the user data into a targeted user and training users.
- 2) Predict the next exhibit and add the predicted exhibit into a set of predicted exhibits (ignoring duplicate predictions).
- 3) Update the targeted user's present location to the next actually visited exhibit.
- 4) Repeat the above two steps until the targeted user's present location becomes the exit.
- 5) Calculate the precision, recall, and F-measure [9] by comparing the set of predicted exhibits to the set of actually visited exhibits.
- 6) Repeat the above process for all users with each becoming a targeted user.
- 7) Average each performance index over all users.

<sup>1</sup> <http://web.mit.edu/museum/>

*PV* is described as follows:

- 1) Divide the user data into a targeted user and training users.
- 2) Predict the next exhibit for the targeted user.
- 3) Compare the predicted exhibit with the actually visited exhibit and accumulated CA (with 0 if wrong and 1 otherwise).
- 4) Update the targeted user's present location to the next actually visited exhibit.
- 5) Repeat the above three steps until the targeted user's present location becomes the exit.
- 6) Calculate the targeted user's final CA by accumulated CA / number of predictions
- 7) Repeat the above process for all users with each becoming a targeted user.
- 8) Average the final CA over all users.

In the proposed method, we extract similar users based on the similarity of their stay time. To calculate the similarity, our proposed method requires some movement information. We, therefore, set the place for starting prediction to the third visit place of a targeted user of interest.

#### B. Prediction accuracy with respect to a threshold value

In this experiment, we also compared the prediction accuracy when the threshold value was determined automatically and when we manually varied its value from 0.1 to 0.9 with the interval of 0.1. This is to examine the usefulness of our mechanism for automatic determination of the threshold.

#### C. Relationship between the prediction accuracy and number of similar users

In addition, we examined the usefulness in exploiting similar user information from the relationship of the prediction accuracy and number of similar users in prediction.

#### D. Relationship between the prediction accuracy and number of similar users

We used Transition Mode (TM) [8] as a baseline model for performance comparison. TM is a first order Markov model based on the frequency information on movement transitions. It is represented by

$$p_{TM}(X_{k+1} = i | X_k = i_k) = TM(i_k, i) = \frac{c(i_k, i)}{c(i_k)}, \quad (13)$$

where  $i_k$  is the current position,  $i$  is the prediction candidate point, and  $c(i_k, i)$  represents the number of occurrences of the transition from  $i_k$  to  $i$ . In order to solve the data sparseness problem, we added a flattening constant  $\epsilon$  ( $= 1/|I|$ ) to each frequency count of transition, where  $|I|$  is the number of exhibits.

## V. RESULTS AND DISCUSSIONS

All results reported here were obtained with leave-one-out cross-validation. Tables II and III show the precision, recall, F-measure, and CA of each method (TM, the proposed KNS, and the proposed KNS with Similar Users (KNS-SIM-cosine and KNS-SIM-euclid)). Table IV, V, VI, and VII show them for each threshold of KNS-SIM-cosine and KNS-SIM-euclid. Finally, Table VIII and IX show the number of similar users for each threshold of KNS-SIM.

TABLE II. OV AND PV PERFORMANCE (USER DATA 1)

	Precision	Recall	F-measure	CA
TM	63.25%	57.47%	59.90%	54.38%
KNS	63.25%	57.47%	59.90%	54.38%
KNS-SIM-cosine	63.55%	58.34%	60.58%	55.25%
KNS-SIM-euclid	65.09%	59.31%	61.74%	56.98%

TABLE III. OV AND PV PERFORMANCE (USER DATA 2)

	Precision	Recall	F-measure	CA
TM	68.91%	66.09%	67.33%	66.09%
KNS	68.91%	66.09%	67.33%	66.09%
KNS-SIM(cosine)	69.34%	65.58%	67.22%	65.58%
KNS-SIM(euclid)	71.65%	66.56%	68.49%	66.19%

TABLE IV. OV AND PV PERFORMANCE OF KNS-SIM-COSINE FOR EACH THRESHOLD (USER DATA 1)

Threshold	Precision	Recall	F-measure	CA
0.1	63.25%	57.47%	59.90%	54.38%
0.2	64.08%	59.17%	61.28%	56.08%
0.3	66.01%	61.18%	63.26%	58.09%
0.4	59.89%	54.89%	57.03%	52.09%
0.5	50.82%	43.94%	46.58%	41.31%
0.6	56.46%	49.10%	51.94%	45.06%
0.7	51.14%	44.46%	47.04%	39.93%
0.8	52.25%	42.78%	46.19%	31.94%
0.9	49.19%	33.81%	38.73%	21.59%
Auto Decision(AVG = 0.26)	63.55%	58.34%	60.58%	55.25%

TABLE V. OV AND PV PERFORMANCE OF KNS-SIM-EUCLID FOR EACH THRESHOLD (USER DATA 1)

Threshold	Precision	Recall	F-measure	CA
0.1	63.25%	57.47%	59.90%	54.38%
0.2	64.12%	59.36%	61.40%	56.27%
0.3	63.81%	58.60%	60.80%	54.38%
0.4	65.75%	60.78%	62.86%	56.56%
0.5	63.02%	57.09%	59.57%	52.28%
0.6	64.47%	58.28%	60.89%	54.43%
0.7	62.35%	55.34%	58.30%	50.15%
0.8	59.11%	46.61%	51.00%	36.63%

Threshold	Precision	Recall	F-measure	CA
0.1	63.25%	57.47%	59.90%	54.38%
0.9	52.30%	31.38%	36.46%	22.53%
Auto Decision(AVG = 0.16)	65.09%	59.31%	61.74%	56.98%

TABLE VI. OV AND PV PERFORMANCE OF KNS-SIM-COSINE FOR EACH THRESHOLD (USER DATA 2)

Threshold	Precision	Recall	F-measure	CA
0.1	69.57%	66.75%	67.99%	66.75%
0.2	69.48%	66.66%	67.90%	66.66%
0.3	70.46%	67.65%	68.89%	67.65%
0.4	70.42%	66.81%	68.41%	66.81%
0.5	70.54%	67.22%	68.67%	66.66%
0.6	69.86%	65.20%	67.22%	63.87%
0.7	74.30%	69.02%	71.22%	62.41%
0.8	70.48%	62.73%	65.63%	49.39%
0.9	50.26%	32.86%	38.19%	26.64%
Auto Decision(AVG = 0.32)	69.34%	65.58%	67.22%	65.58%

TABLE VII. OV AND PV PERFORMANCE OF KNS-SIM-EUCLID FOR EACH THRESHOLD (USER DATA 2)

Threshold	Precision	Recall	F-measure	CA
0.1	68.91%	66.09%	67.33%	66.09%
0.2	69.57%	66.75%	67.99%	66.75%
0.3	69.85%	67.04%	68.28%	67.04%
0.4	69.21%	65.98%	67.39%	65.41%
0.5	70.99%	66.64%	68.34%	65.69%
0.6	73.09%	67.63%	69.54%	63.36%
0.7	74.01%	61.30%	65.32%	57.15%
0.8	76.31%	57.53%	62.28%	53.52%
0.9	51.52%	32.90%	35.61%	18.18%
Auto Decision(AVG = 0.32)	71.65%	66.56%	68.49%	66.19%

TABLE VIII. THRESHOLD VALUES AND NUMBER OF SIMILAR USERS (USER DATA 1)

Threshold	Cosine	Euclid
0.1	36.54	41.79
0.2	30.71	39.6
0.3	25.79	37.67
0.4	20.93	35.3
0.5	15.71	32.44
0.6	11.78	28.37
0.7	8.12	20.16
0.8	5.06	9.7
0.9	2.48	2.54
Auto Decision	27.5(AVG=0.26)	38.77(AVG=0.16)

TABLE IX. THRESHOLD VALUES AND NUMBER OF SIMILAR USERS (USER DATA 2)

Threshold	Cosine	Euclid
0.1	33.62	42.23
0.2	27.87	40.29
0.3	21.84	37.05
0.4	16.81	32.94
0.5	12.50	27.37
0.6	8.54	18.87
0.7	5.08	10.25
0.8	2.96	5.58
0.9	1.28	1.84
Auto Decision	21.26 (AVG=0.32)	28.39(AVG=0.36)

We first discuss the results in the Tables II and III. In these tables, we can see that the results of KNS-SIM-euclid are the highest for all performance indexes while TM and KNS are of the same prediction performance. This indicates that the information on similar users is more important than other non-similar users for KNS. In addition, this implies that our mechanisms for extracting similar users are effective, i.e., the stay-time similarity (Eqns. 9 and 10) and the algorithm for automatically deciding the threshold (Algorithm 3).

We now discuss the results in Table IV, V, VI, and VII. From these results, we can see that the automatic threshold determination approach outperforms the manual approach for many cases. Since the optimal threshold value is problem dependent, the automatic approach can thus reduce the burden to manually search for such a value and at the same time can provide a promising value automatically.

Finally, we discuss the results in Table VIII, IX. These tables show that the average number of similar users when threshold was automatically determined. These numbers are less than 44, the number of training users used in TM or KNS – please recall that we used leave-one-out cross-validation. As a result, we can suggest that an increase in the size of the training users does not necessarily lead to improvement in prediction accuracy. What is more important is the suitable amount of similar users for performing maximum likelihood estimation, which could be extracted from the training data set by the automatic threshold determination mechanism given in this paper.

## VI. CONCLUSIONS

In this paper, we proposed a method that uses the information on similar users in the statistical language-model

smoothing technique KNS for user pathway prediction. The proposed method, which has also a couple of original mechanisms for extracting similar users from the training data set, outperformed the other methods as shown in the results from the conducted experiment. This indicates that our idea to exploit only similar users in KNS is effective. As our future work for further improving the prediction accuracy, we plan to focus on attribute information such as age, sex, and hobbies in addition to the stay time and find how to combine them for this task.

## ACKNOWLEDGMENT

This work is supported in part by Grant-in-Aid for Scientific Research (C), No. 23500192, the Japan Society for Promotion of Science.

## REFERENCES

- [1] R. Tesoriero, J. A. Gallud, M. Lozano, V. M. R. Penichet: A Locationaware System using RFID and Mobile Devices for Art Museums. Fourth International Conference on Autonomic and Autonomous Systems, pp.76- 81(2008)
- [2] A Survey of Context-Aware Mobile Computing Research.(2005) [https://mmlab.snu.ac.kr/courses/2005\\_advanced\\_internet/handout/ppt/36%20%20context\\_aware.pdf](https://mmlab.snu.ac.kr/courses/2005_advanced_internet/handout/ppt/36%20%20context_aware.pdf)
- [3] Fabian Bohnert and Ingrid Zukerman: Personalised Pathway Prediction. User Modeling, Adaptation, and Personalization Lecture Notes in Computer Science Volume 6075, pp. 363–368 (2010)
- [4] Chen Cheng, Haiqin Yang, Irwin King, Michael R. Lyu: Fused Matrix Factorization with Geographical and Social Influence in Location-Based Social Networks. Twenty-Sixth AAAI Conference (2012)
- [5] Stanley F. Chen and Joshua Goodman: An empirical study of smoothing techniques for language modeling. Computer Speech and Language 13, pp. 359–394 (1999)
- [6] Flavia Sparacino: Sto(ry)chastics: a Bayesian Network Architecture for User Modeling and Computational Storytelling for Interactive Spaces. UbiComp 2003: Ubiquitous Computing Lecture Notes in Computer Science, Volume 2864, pp. 54–72 (2003)
- [7] Kingkarn Sookhanaphibarn and Ruck Thawonmas: Exhibition-Area Segmentation Using Eigenvectors. International Journal of Digital Content Technology and its Applications, Volume 7, Number 2, pp. 533–540 (2013)
- [8] Fabian Bohnert, Ingrid Zukerman, Shlomo Berkovskya, Timothy Baldwin, and Liz Sonenberg: Using Interest and Transition Models to Predict Visitor Locations in Museums. AI Communications, Volume 21, Number 2-3, pp. 195–202 (2008)
- [9] John Makhoul, Francis Kubala, Richard Schwartz, Ralph Weischedel: Performance measures for information extraction. DARPA Broadcast News Workshop, pp. 249–253 (1999)

[Top](#)

[Program Guide](#)

[author Index](#)

[iCAST 2013  
Technical Program](#)

[UMEDIA 2013  
Technical Program](#)